FUNDAÇÃO DE ENSINO “EURIPEDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO “EURIPEDES DE MARILIA” – UNIVEM
TRABALHO DE CONCLUSÃO DE CURSO

**RAPHAEL GIOVANINI DA SILVA**

# SADI – IMPLEMENTAÇÃO DOS NÍVEIS DE SEGURANÇA E ACESSO REMOTO

MARILIA
2005

RAPHAEL GIOVANINI DA SILVA

# SADI – IMPLEMENTAÇÃO DOS NÍVEIS DE SEGURANÇA E ACESSO REMOTO

Monografia apresentada ao Curso de Ciência da Computação, da Fundação de Ensino "Eurípides Soares da Rocha", Mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, para obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Prof. Dr. Antonio Carlos Sementille

MARILIA
2005

RAPHAEL GIOVANINI DA SILVA

# “SADI – IMPLEMENTAÇÃO DOS NÍVEIS DE SEGURANÇA E ACESSO REMOTO”

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 8,00 ( oito )

Orientador: Antonio Carlos Sementille _____

1°. Examinador: José Remo Ferreira Brega _____

2°. Examinador: Ildeberto Aparecido Rodello _____

Marília, 07 de dezembro de 2005.

***Dedico este trabalho aos meus pais, Sr. Jair Lopes da Silva e Sra. Margaret Elaine Giovanini da Silva, por vocês tenho muita gratidão por ter tudo o que tenho em minha vida.***

# AGRADECIMENTOS

Aos meus pais, por tudo que me ensinaram e todo amor e carinho e pela paciência que tiveram comigo durante esses 21 anos de vida.

A minha irmã Renata, que sempre esteve ao meu lado nas horas que precisei.

Ao meu grande amigo Marcel (acreano), que infelizmente hoje não estuda mais comigo, mais é o meu maior amigo que conquistei e nos ajudamos durante esses 4 anos.

Antonio João, grande amigo de infância, que está longe agora, mas também está se formando e depois disso iremos sumir de férias pra praia.

Ao meu orientador e professor, o grande Sementille, pela paciência pelos ensinamentos, e acima de tudo ser um grande homem e um grande professor.

Ao meu professor Remo, o cara da camisa rosa, pelas enquêtes feitas, os "pogramas" desenvolvidos, e ao "GIANEQUINI".

O Beto achou que eu ia esquecer dele, mas não, eu acho q mostrei pra ele que consigo surpreender a todos. E aí, vamo toca ou não?Não chama o remo pra tocar pandeiro não.

Aos amores que se foram e só me deixaram atordoado.

SILVA, Raphael Giovanini. SADI – Implementação dos níveis de segurança e acesso remoto. 2003. 62f. Relatório de TCC – Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Euripedes de Marília, Fundação de Ensino Euripedes Soares da Rocha, Marília, 2005.

## RESUMO

Atualmente os ataques às redes de computadores são uma verdadeira ameaça a segurança das informações. Sendo assim, foi desenvolvida uma ferramenta denominada SADI (Sistema Adaptável de Detecção de Intrusão), com a finalidade de detectar e classificar ações de ataque que podem resultar em acesso não autorizado na rede de uma empresa ou no microcomputador de um usuário. Neste contexto, o presente trabalho objetivou a criação de mais um módulo do SADI, o qual consiste no gerenciamento e configuração remota, além do aperfeiçoamento do módulo anterior, em que foi adicionado um módulo de análise do nível de segurança.

**Palavras-chave:** Redes de Computadores, Segurança, PHP, MySQL, Winpcap.

SILVA, Raphael Giovanini. SADI – Implementação dos níveis de segurança e acesso remoto. 2003. 62f. Relatório de TCC – Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Euripedes de Marília, Fundação de Ensino Euripedes Soares da Rocha, Marília, 2005.

## ABSTRACT

Lately the attacks against computers network are a real threat to information security. Because of that, a tool called SADI was developed to detect and classify actions that can result on a not-allowed access of a company or personal computer. In this context, this paper work looked forward to create a new segment of SADI that will make remote management and configuration of the system and an upgrade of the last segment that was added to a segment of analysis of security level.

**Keywords:** Computer Networks, Security, PHP, MySQL, Winpcap.

# LISTA DE ABREVEATURAS

| | |
|---|---|
| DARPA | Defense Advanced Research Project Agency |
| DHCP | Dynamic Host Control Protocol- |
| DNS | Domain Name Service |
| DoD | Departamento de Defesa dos Estados Unidos |
| DoS | Denial of Service |
| IDS | Intrusion Detection System |
| IP | Internet Protocol |
| NS | Nível de Segurança. |
| OSI | Open System Interconnection |
| SADI | Sistema Adaptável de Detecção de Intrusos |
| SNMP | Simple Network Management Protoco |
| TCP | Transport Control Protocol |
| TFTP | Trivial File Transfer Protocol |
| UDP | User Datagram Protocol |

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABELAS

# SUMÁRIO

# CAPÍTULO 1 - INTRODUÇÃO

Com o grande crescimento de incidentes de segurança, aumenta o número de ferramentas que detectam intrusos, e nesse projeto será apresentado um projeto para detecção de ações intrusivas, denominado SADI (Sistema Adaptável Detecção de Intrusos), algumas definições e descrição do projeto e seus aprimoramentos.

Seus aprimoramentos, como o módulo de análise de nível de segurança, que classifica a intrusão, e armazena em um banco de dados a mesma, e um módulo de acesso remoto, que contém todas as informações para análise e mais as que foram capturadas, com uma interface simples e totalmente voltada ao usuário, uma ferramenta poderosa.

## 1.1 O problema e a importância da detecção de intrusão

Com o crescimento do número de usuários na rede de computadores, aumentou-se também à preocupação com a segurança dos dados que trafegam na rede e toda a informação.

Intrusão (ou ataque) é todo o acesso não autorizado de indivíduos (hacker), e/ou ações que possam comprometer a integridade, confidencialidade ou disponibilidade de um recurso computacional.

A detecção de intrusos num sistema pode ser definida em duas categorias: detecção de intrusão por anomalia e detecção por abuso. A detecção por anomalia é baseada na determinação de comportamento anômalo no uso de recursos do sistema. Por exemplo, se normalmente um determinado usuário *A* do departamento de vendas de uma empresa somente usa sua conexão de rede de segunda a sexta-feira entre 8 e 16 horas, uma atividade noturna na conta deste usuário, é anormal, e pode significar uma intrusão. A detecção por anomalia tenta

quantificar o comportamento usual ou aceitável, enquanto tenta indicar outros comportamentos irregulares como sendo potencialmente intrusivos. (DIAS et al, 2004)

## 1.2 Tipos de Ataques

### 1.2.1 Intrusão

A intrusão é definida como um conjunto qualquer de ações que viola a política de segurança de um sistema, comprometendo sua integridade, confidencialidade e disponibilidade do mesmo.

Existem três espécies de ataques que são mais conhecidos por ocorrerem em ambientes de redes: penetração de sistemas, varreduras, e negativas de serviço (*DoS – Denial of Service*), (SILVA, 2002, p. 55)

**Ataques de penetração** consistem na aquisição ou alteração não autorizada dos privilégios, recursos ou dados do sistema. Um ataque de penetração pode ganhar controle de um sistema ao explorar uma variedade de falhas de software como as encontradas em algumas implementações antigas dos servidores *telnet*.

**Varreduras** é o tipo de ataque ocorrem quando são realizadas prospecções em uma rede enviando a eles diferentes tipos de pacotes. As respostas servem para aprender muitas características como à topologia da rede dos alvos e suas configurações de software.

**Ataques de negação de Serviço** é um tipo de ataque visa diminuição de desempenho ou interrupção de serviço de rede e do sistema. Existem três principais tipos e ataques de negação de serviço:

- **Exploração de falhas:** esse tipo de ataque explora as vulnerabilidades no software e que causam falhas no processamento ou extinguem os recursos do mesmo.

- **Flooding:** esse tipo de ataque envia a um sistema informações a mais que sua capacidade. Ele causa uma monopolização da conexão da rede alvo, bloqueando assim qualquer tipo de uso deste recurso.

- **Ataques de negativa de serviço distribuído:** é um tipo de ataque com uma estrutura previamente montada, onde diversas máquinas atacam baseadas em *flooding* uma vítima alvo. Tem por finalidade ter um número de requisições de conexão ao servidor maior.

Uma cena típica de ataque: o primeiro passo é a procura de vulnerabilidades específicas, de forma a estabelecer um banco de dados de endereços que contém candidatos em potencial para serem atacados. Logo após, deve-se tentar explorar as falhas encontradas.

Existem várias abordagens para classificar um ataque, outra forma é a seguinte, se baseando no agrupamento das intrusões dependendo do seu efeito final e método de ação:

- Tentativa de invasão (*Attempted break-in*): é determinado também através de perfis de tentativas de violação dos mecanismos de validação e contenção com comportamento atípico.

- Ataque dissimulado (*Masquerade Attack*): é semelhante ao de tentativa de invasão, mas com determinação mais sensível.

- Penetração do controle de segurança do sistema (*Penetration*): usualmente detectado pela monitoração de padrões específicos de atividade.

- Escoamento (*Leakage*): é detectado pelo uso atípico de recursos de entrada e saída e que não se encaixa nos casos anteriores.

- Comprometimento dos recursos (*Denial of Service*): detectado por uso atípico dos recursos do sistema, e que os leva a se tornarem indisponíveis.

- Uso Malicioso (*Malicious use*): é relacionado a usuários legítimos do sistema que abusam de seus privilégios tentando violações de mecanismos de bloqueios, ou uso de privilégios especiais. (SILVA, 2002, p. 55)

A detecção de intrusos pode ser divida em duas categorias: detecção de intrusão por anomalia e detecção por abuso.

### 1.2.2 Detecção de Anomalia

O conceito principal de anomalia é quando a atividade intrusiva é um subconjunto de atividade anormal. É um conceito simples, sabendo-se que o atacante externo tem uma grande chance de possuir um comportamento anormal dentro de uma conta de um usuário do qual ele não conhece o padrão de atividades intrusivas. Há quatro casos possíveis de detecção de anomalia, em que cada uma a probabilidade diferente de zero de ocorrer.

-Intrusivo e anômalo – é a atividade intrusiva e é apontada como anômala, são os chamados verdadeiros positivos.

- Não Intrusivo e não anômalo – é uma atividade não intrusiva e é apontada como não anômala, são chamados de verdadeiros negativos.

- Intrusivo, mas anômalo – é uma atividade intrusiva, em que ocorre uma falha na sua detecção por ela não ser anômala são chamados de falsos positivos.

- Não intrusivo, mas anômalo – é uma atividade não intrusiva, e que o sistema aponta de forma incorreta ser intrusiva, pelo fato dela ser anormal, são chamados de falsos positivos.

### 1.2.3 Detecção por Abuso

A característica principal na detecção por abuso é que há ataques que podem ser precisamente codificados, de maneira a capturar e registrar arranjos e variações acerca de atividades que exploram a mesma vulnerabilidade. Entretanto, na prática, nem todas as maneiras possíveis de se efetuar uma intrusão podem ser eficientemente capturadas e

codificadas. A principal limitação desse método é a busca por vulnerabilidades conhecidas, o que não pode ser de muita utilidade na detecção de intrusões futuras. Outra limitação deste método tem a ver com considerações práticas sobre o que é auditorado. Por exemplo, as práticas convencionais de auditoria não registram mudanças nas variáveis dos processos, devido ao impacto que isso pode causar no desempenho do sistema e devido às exigências de disco para armazenar os dados auditorados. Se uma intrusão só puder ser identificada a partir daqueles dados, há um comprometimento do método, ainda que alguns dados possam ser inferidos a partir de outras condições do sistema (CANSIAN, 1997, p. 9).

Por outro ponto de vista, os métodos de detecção por abuso, são preferidos em muitos casos devido ao baixo custo computacional e ao pequeno comprometimento do desempenho. Além disso, a maioria das tentativas de intrusão ocorre, a partir de padrões bem definidos de comportamento, e que freqüentemente são de conhecimento dos administradores do sistema.

## 1.3 Descrição dos Capítulos

Aqui estão as descrições dos próximos capítulos, sendo que o presente capítulo faz uma breve introdução sobre o projeto e o conteúdo envolvido com o mesmo.

O Capítulo 2 contém os conceitos relacionados ao projeto, onde serão apresentadas informações sobre o protocolo TCP/IP, segurança de rede, detecção de intrusão, assinaturas de ataque,

O Capítulo 3 contém a caracterização dos sistemas de detecção de intrusos, e os principais sistemas encontrados.

O Capítulo 4 contém o sistema SADI, sua estrutura, seus módulos.

O Capítulo 5 contém a descrição do trabalho realizado e seus objetivos, o novo módulo de acesso remoto, o módulo de análise do nível de segurança, que foram acrescentados ao SADI.

No Capítulo 6, são apresentados os resultados dos testes realizados no sistema SADI, em um ambiente experimental.

No Capítulo 7 são feitas as conclusões sobre o projeto

No Capítulo 8 são apresentadas propostas para futuras implementações e as dificuldades encontradas.

# CAPÍTULO 2 - Fundamentos Teóricos

## 2.1 Redes de computadores – Arquitetura TCP/IP

TCP/IP é o nome do conjunto de protocolos utilizados pela Internet. Este conjunto de protocolos foi desenvolvida pela DARPA (Defense Advanced Research Project Agency) no DoD (Departamento de Defesa dos Estados Unidos).

Foi desenvolvido para que os computadores compartilharem recursos numa rede. Toda a família de protocolos inclui um conjunto de padrões que especificam os detalhes de como comunicar computadores, assim como também convenções para interconectar redes e roteamento do tráfego.

A arquitetura básica do protocolo TCP/IP é mostrada na Figura 2.1, e fazendo relação com o modelo de referência OSI correspondem às camadas da Rede, Transporte e Aplicações.

Observando na Figura 2.1, na camada de rede encontra-se o protocolo IP (*Internet Protocol*); na camada de transporte estão dois protocolos, um que oferece serviços sem conexão, que é o protocolo UDP (*User Datagram Protocol*) e um outro que oferece serviços orientados a conexão, protocolo TCP (*Transport Control Protocol*). Na camada de aplicações existe uma variedade de protocolos de aplicação, como SMTP, TELNET, FTP e NSF entre outros.

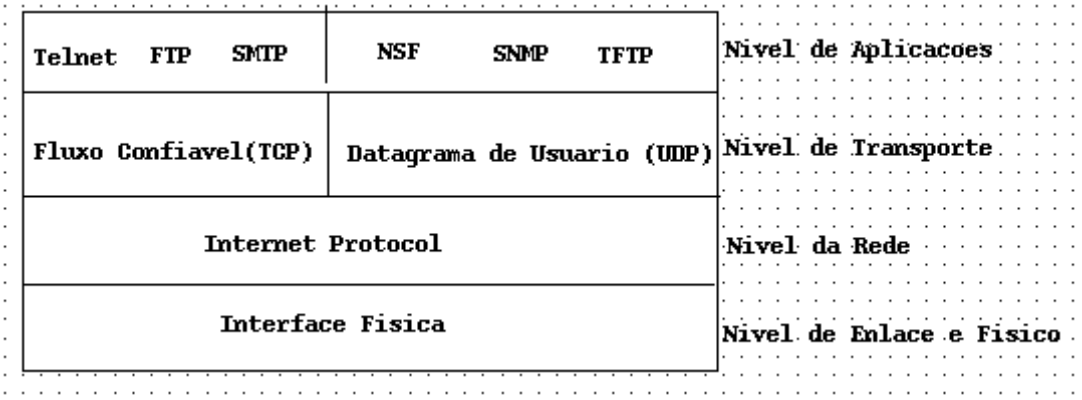


Figura 2.1: Arquitetura TCP/IP

No nível de rede se mostra o protocolo que oferece serviços sem conexão sobre o qual os serviços dos dois níveis superiores trabalham.

Oficialmente este conjunto de protocolos é chamado, Protocolo Internet TCP/IP, mas que é referenciada só como TCP/IP, por causa de seus dois protocolos de maior importância (TCP: Transport Control Protocol e IP: Internet Protocol).

### 2.1.1 Endereços IP

Um sistema provê um serviço de comunicação universal se ele permite a qualquer host se comunicar com outro qualquer. Para que um sistema preste serviços de comunicação universalmente é necessário estabelecer um método de identificar os computadores que seja aceito globalmente. Na Internet se escolhe identificar os computadores através de endereços binários.

Na Internet a cada computador é associado um endereço inteiro de 32 bits, chamado endereço IP. O importante no esquema de endereços internet é que os inteiros são cuidadosamente escolhidos para fazer o roteamento eficiente. Especificamente um endereço

IP define o identificador da rede ao qual o host está conectado e também a identificação de um único computador nessa rede.

Conceitualmente cada endereço é um par (netid, hostid), onde netid identifica a rede, e hostid identifica um computador nessa rede. Na prática cada endereço IP deve ter um Formato específico. Na Figura 2.2 é apresentado o formato dos endereços IP.

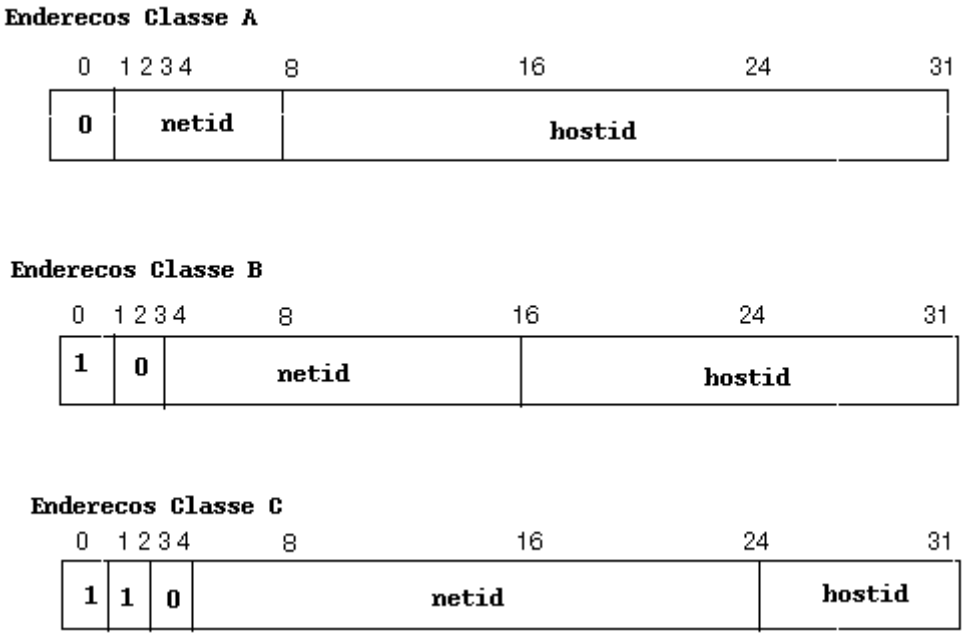


Figura 2.2: Endereços IP

## 2.1.2 Formato do Datagrama IP

O datagrama IP é *a unidade básica de dados* no nível IP. Um datagrama está dividido em duas áreas, uma área de cabeçalho e outra de dados.

O cabeçalho contém toda a informação necessária que identificam o conteúdo do datagrama.

Na área de dados está encapsulado o pacote do nível superior, ou seja, um pacote TCP ou UDP.

O formato do datagrama IP é representado na figura 2.3:

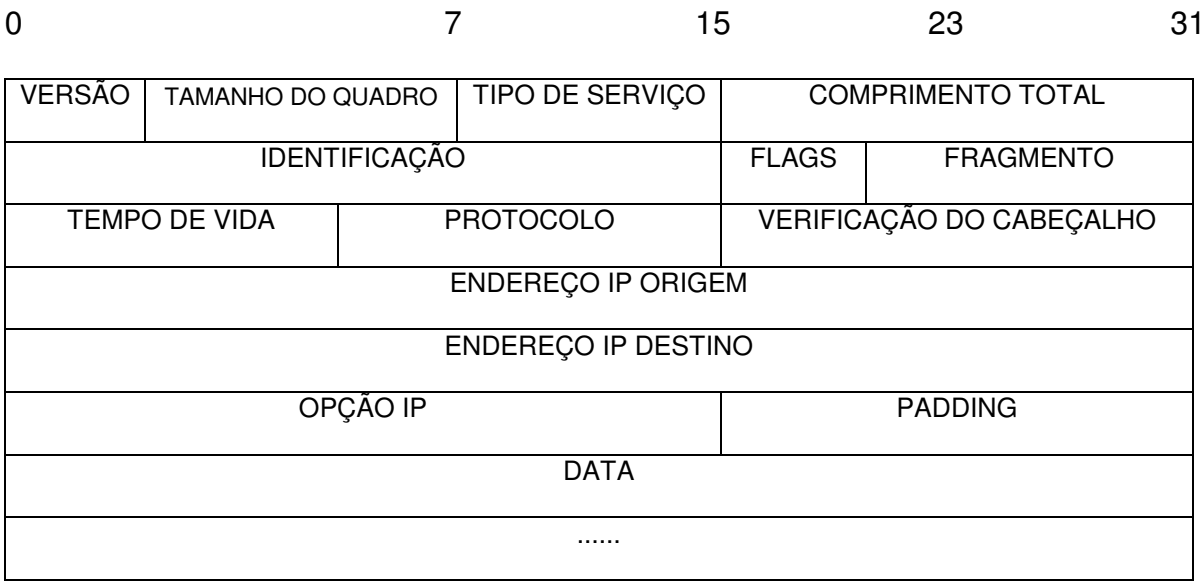


Figura 2.3: Cabeçalho *IP*

Onde:

**VERSÃO**: versão do protocolo que foi usada para criar o datagrama (4bits)

**TAMANHO DO QUADRO**: comprimento do cabeçalho, medido em palavras de 32 bits (4 bits)

**TIPO DE SERVIÇO**: este campo especifica como o datagrama e pode ser manejado e dividido em cinco subcomandos:

*Precedence*:(3 bits) indica precedência de datagramas com valores desde 0 (precedência normal) até 7 (controle da rede), com estes bits permite-se ao transmissor indicar a importância de cada datagrama que ele está enviando.

*Bits D,T,R*: indicam o tipo de transporte que o datagrama deseja, Baixo Retardo(D), Alta Capacidade de Processamento(T) e Alta Confiabilidade(R).

Não é possível que estes tipos de serviços sempre sejam oferecidos, já que dependem das condições físicas da rede.

**COMPRIMENTO TOTAL**: informa o comprimento do datagrama medido em bytes, e inclui cabeçalho e dados.

**IDENTIFICAÇÃO**: possibilita ao *host* determinar a que datagrama pertence um fragmento recém-chegado (todos os fragmentos de um datagrama possuem o mesmo valor).

**FLAGS**: é composto de um bit não utilizado seguido por dois bits, *DF* e *MF*. O *DF* significa *Don't Fragment* e indica que os *gateways* não devem fragmentar este datagrama (por incapacidade do destino juntar novamente os fragmentos). *MF* significa *More Fragments*, e é utilizado como dupla verificação do campo *TOTAL LENGTH*, sendo que todos os fragmentos, menos o último possuem este bit setado.

**FRAGMENTO**: informa a que posição no datagrama atual pertence o fragmento.

**TEMPO DE VIDA**: especifica o tempo em segundos que o datagrama está permitido a permanecer no sistema Internet. Gateways e hosts que processam o datagrama devem decrementar o campo de Tempo de Vida cada vez que um datagrama passa por eles e devem removê-lo quando seu tempo expirar.

**PROTOCOLO**: especifica qual protocolo de alto nível foi usado para criar a mensagem que está sendo transportada na área de dados do datagrama.

**VERIFICAÇÃO DO CAEÇALHO**: assegura integridade dos valores do cabeçalho.

**ENDEREÇO IP ORIGEM E DESTINO**: especifica o endereço IP de 32 bits do remetente e receptor.

**OPÇÕES IP**: é um campo opcional. Este campo varia em comprimento dependendo de quais opções estão sendo usadas. Algumas opções são de um byte, e neste caso este campo é chamado de Option Code e está dividido em três campos.

*COPY*:(1 bit) controla a forma em que o gateway trata as opções durante a fragmentação:

**1:** a opção deve ser copiada em todos os fragmentos

**0:** a opção deve ser copiada somente no primeiro fragmento.

*CLASS*(2 bits): especifica a classe geral da opção.

***OPTION NUMBER()*:** especifica uma opção na classe determinada no campo CLASS.

**Nível de Enlace de Dados:** os protocolos desse nível estão envolvidos com as controladoras de comunicação, seus microprocessadores e seus *buffers*. Dentro do conjunto TCP/IP existem dois protocolos denominados ARP (*Address Resolution Protocol*) e RARP (*Reverse ARP*) , que situam-se entre os níveis de Rede e de Enlace de Dados, para tratar o endereçamento quanto ao meio físico.

**ARP**: é o protocolo que resolve os endereços físicos, mapeia o endereço IP conhecido de 32 bits para o endereço físico na interface de rede. Por exemplo, em nosso caso, o IP é mapeado para um endereço físico e*thernet*, composto de 48 bits.

**RARP**: é o protocolo reverso do ARP, que resolve os endereços IP. Mapeia o endereço físico conhecido (no nosso exemplo, *ethernet*) para um endereço IP.

**Nível Físico:** é o nível que tem contato direto com o *hardware* da máquina, tratando os sinais eletrônicos. Transmite e recebe os bits que dão origem à estrutura conhecida como pacote (CANSIAN, 1997, p. 31).

## 2.2 Protocolo TCP (Transport Control Protocol)

TCP é um protocolo presente no nível de transporte. E é um protocolo orientado a conexão, o que indica que neste nível vão ser solucionados todos os problemas de erros que não forem solucionados no nível IP, dado que este último é um protocolo sem conexão. Alguns dos problemas com os que TCP deve tratar são os pacotes perdidos ou destruídos por erros de transmissão e a expedição de pacotes fora de ordem ou duplicados.

O TCP especifica o formato dos pacotes de dados e de reconhecimentos que dois computadores trocam para realizar uma transferência confiável, assim como os procedimentos que os computadores usam para assegurar que os dados cheguem corretamente. Entre estes procedimentos estão distinguir entre múltiplos destinos numa máquina determinada e fazer recuperação de erros, tais como pacotes perdidos ou duplicados.

TCP permite que múltiplos programas de aplicação numa determinada máquina se comuniquem concorrentemente. TCP se encarrega de demultiplexar o tráfego TCP entrante entre os programas de aplicação.

TCP usa número de portas para identificar o último destino numa máquina. A cada porta é associado um número inteiro pequeno para identificá-lo.

TCP foi construído sobre a abstração de conexão, na qual os objetos a serem identificados são conexões de circuitos virtuais e não portas individuais. As conexões são identificadas por um par de "endpoints". Uma conexão consiste de um circuito virtual entre dois programas de aplicações, então se pode assumir um programa de aplicação como a conexão entre os endpoints, mas isto não é certo, TCP define um endpoint como um par de inteiros (host,port), onde *host* é o endereço IP para um computador e *Port* é uma porta TCP nesse computador.

Exemplo: 128.10.2.3.16 especifica a porta TCP número 16 na máquina como o endereço IP 128.10.2.3.

O TCP identifica uma conexão por um par de endpoints, um número de porta pode ser compartilhado por múltiplas conexões na mesma máquina.

O TCP vê o fluxo de dados como uma seqüência de bytes, que ele divide em segmentos para a transmissão. Sendo assim cada segmento viaja através da Internet com um único datagrama IP.

TCP usa um mecanismo de janela deslizante para resolver dois problemas importantes, que é a Transmissão eficiente e o Controle de fluxo.

### 2.2.1 Formato do Segmento TCP

A unidade de transferência entre o software TCP de duas máquinas é chamada Segmento. Os segmentos são trocados para estabelecer conexões, transferir dados, enviar reconhecimentos e fechar conexões. Dado que TCP usa a técnica de *Piggybacking*, um reconhecimento viajando de uma máquina A a B pode ir no mesmo segmento de dados que estão sendo enviados de A a B, embora o reconhecimento refere-se a dados enviados da máquina B a A.

O formato do segmento TCP será apresentado a seguir (Figura 2.4):

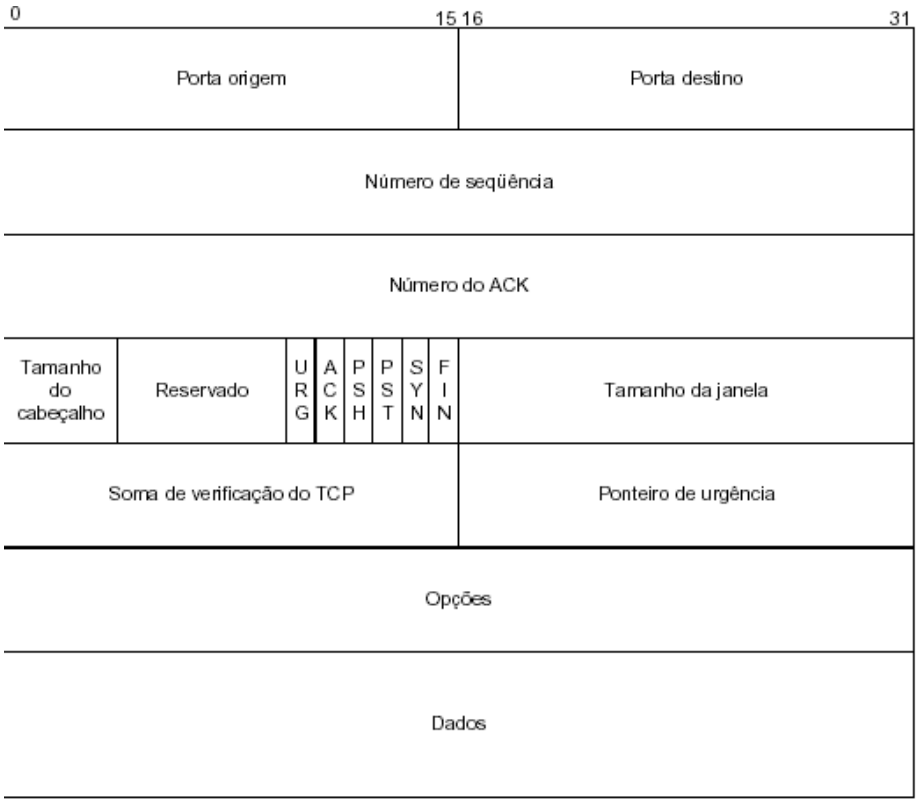


Figura 2.4: *Formato do segmento TCP*

***Porta Fonte e Destino***: estes campos no cabeçalho TCP contêm os números de portas TCP que identificam os programas de aplicação dos extremos de uma conexão.

***Número de seqüência*** *(32 bits)*: identifica a posição no fluxo de bytes do segmento enviado pelo transmissor. O número de seqüência refere-se ao fluxo de dados que vai à mesma direção do segmento.

***Número de Reconhecimento*** *(32 bits)*: este campo identifica a posição do byte mais alto (ou último byte) que código fonte recebeu. O número de reconhecimento refere-se ao fluxo de dados na direção contrária ao segmento. Os reconhecimentos sempre especificam o número do próximo byte que o receptor espera receber.

***Offset***: contém um inteiro que especifica o início da porção de dados do segmento. Este campo é necessário já que o campo *Options* varia em comprimento dependendo de quais opções tenham sido incluídas. De modo que o tamanho do cabeçalho TCP varia dependendo das opções selecionadas.

***RES*****:** reservado para uso futuro.

***CODE*** *(6 bits)*: determina o propósito e conteúdo do segmento, codificado assim:

Bits de esquerda a Direita Significado:

URG - Campo de ponteiro Urgente é válido

ACK - Campo de Reconhecimento é válido

PSH - Este segmento solicita um PUSH

RST - Reset da conexão

SYN - Sincroniza numeros de sequências

FIN - O transmissor chega ao fim do fluxo de bytes.

***TAMANHO DA JANELA***: através deste campo o software TCP indica quantos dados ele tem capacidade de receber em seu buffer.

***URGENT POINTER***: TCP através deste campo permite que o transmissor especifique que alguns dados são urgentes, isto significa que os dados serão expedidos tão rápido quanto seja possível.

***OPTIONS***: o software TCP usa este campo para se comunicar com o software do outro extremo da conexão.

***CHECKSUM***: é usado para verificar a integridade tanto do cabeçalho como dos dados do segmento TCP.

## 2.3 Protocolo UDP (User Datagram Protocol)

UDP provê um serviço sem conexão não confiável, usando IP para transportar mensagens entre duas máquinas.

Este protocolo, igualmente o TCP, provê um mecanismo que o transmissor usa para distinguir entre múltiplos receptores numa mesma máquina.

Cada datagrama UDP é formado por um cabeçalho UDP e uma área de dados.

O formato do cabeçalho UDP está dividido em quatro campos de 16 bits. O segmento UDP é ilustrado na Figura 2.5

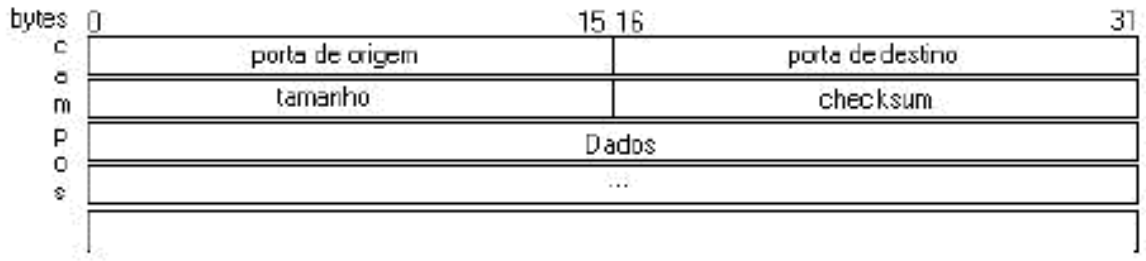


Figura 2.5: Formato do segmento UDP

Onde:

**Porta de Origem e Porta Destino:** estes campos contêm os números de portas fonte e destino do protocolo UDP. A porta fonte é opcional, quando é usada ela especifica a porta a qual uma resposta poderia ser enviada, se não é usada contém zeros.

**Tamanho:** contém um contador de bytes no datagrama UDP. O valor mínimo é oito, sendo este só o comprimento do cabeçalho.

**Checksum:** Este campo é opcional. Um valor de zero indica que o checksum não é computado.

Como dito anteriormente, o protocolo UDP é simples se comparado ao TCP, então somente alguns protocolos utilizam o UDP para transporte de dados que são: o TFTP (*Trivial File Transfer Protocol*), SNMP (*Simple Network Management Protocol*), DHCP (*Dynamic Host Control Protocol*), DNS (*Domain Name Service*).

**TFTP -** Este protocolo é semelhante ao FTP, porém não há confirmação de recebimento pelo destino ou reenvio.

**SNMP -** É utilizado para configuração de dispositivos, como roteadores ou *switches* e permite que estes enviem o seu *status*. As últimas versões do SNMP podem fazer criptografia md5, porém a maioria ainda usa versões antigas que passa o password em formato de texto.

**DHCP -** É utilizado em redes que sofrem constantes alterações na topologia e o administrador não pode verificar o IP (*Internet Protocol*) de cada máquina devido a enorme quantidade, então o roteador distribui IPs automaticamente para as estações. Como esta atribuição é feita com a utilização do UDP, caso haja algum problema o usuário terá que pedir o reenvio ou reiniciar a máquina. O único problema técnico deste protocolo é que como os IPs

são atribuídos aleatoriamente, fica mais difícil para o administrador ter controle sobre o que cada *host* está fazendo.

**DNS -** Um tradutor dos nomes na rede, na qual cada IP pode ser correspondido com um nome. Neste caso, imaginemos que um usuário esteja acessando a internet e deseja ir para outra página. Ele digita o endereço no campo apropriado e entra. Se a página, por acaso, não abrir por não ter reconhecido o endereço, o problema poderá ter sido no envio ou resposta do servidor de nomes utilizando o UDP, e então o usuário tentará de novo acessar a página e provavelmente conseguirá. Agora, imagine que isto fosse feito com o TCP, provavelmente esta falha não ocorreria, porém o tempo gasto para o computador saber qual IP se refere àquele nome seria inimaginável para as necessidades atuais.

# CAPÍTULO 3 – SISTEMAS DE DETECÇÃO DE INTRUSÃO

## 3.1 Caracterização

Um sistema de detecção de intrusão (IDS – *Intrusion Detection System*) é um software, que tem a função de detectar atividades maliciosas ou anômalas, monitorando a rede continuamente ou servidor capturando todos os pacotes que por ali passam.

## 3.2 SNORT

O SNORT é uma ferramenta muito simples uso e com recursos extraordinários, eis alguns dos motivos que me levou a escolhe-l como ser um software livre, de fácil, tem versões tanto pra sistemas operacionais Windows como Linux, ser de fácil manuseio.

O SNORT é que o mesmo permite atualização automática de suas assinaturas de ataque via internet.

Com o SNORT, o próprio usuário personaliza sua regra de ataques e escolhe-os da forma que melhor se adapta.

É possível escolher um ou todos dentre os mais diversos tipos de ataques, as portas, o tipo de protocolo entre outros, após isso baste clicar em create this ruleset e pronto, está criado a lista com as regras que serão usadas para determinar os tipos de ataque.

### 3.2.1 Conselhos para instalação

É aconselhável que antes da instalação e configuração do SNORT, se tenha alguns tipos de precauções para que tudo funcione perfeitamente. Como todos sabem, uma

ferramenta IDS serve para que possamos ter controle sobre o que está acontecendo em nossa rede, para em casos como sucessivos portscans ou até mesmo uma invasão, possamos fazer uma denuncia aos órgãos responsáveis, porém, o que ninguém sabe, ou melhor, a grande maioria se esquece é que para que a denuncia seja feita, é necessário que alem das provas (logs) precisamos estar com o relógio do computador onde está instalada a ferramenta IDS perfeitamente sincronizado, ou seja, a hora tem que estar exata para que não haja inconsistência dos dados a serem analisados.

Um outro tipo de precaução que também deve ser tomada é quanto ao sistema onde vai ser instalada a ferramenta IDS, e para evitar que seja instalado em um sistema que esteja com vulnerabilidades ou até mesmo algum rootkit instalado. Para solucionar o problema é necessário que o sistema seja reinstalado e que sejam baixados todos os pacotes contendo as correções para vulnerabilidades, isto é feito facilmente consultando o site do fabricante do seu sistema operacional.

## 3.3 Instalação e configuração

Após ter o programa e instalá-lo, precisamos configurá-lo para que ele entenda quais arquivos de assinatura devem ser chamados ou quais tipos de ataque devem ser monitorados. Não vou entrar em detalhes da instalação por existir diversas versões de Sistemas Operacionais, portanto vou me limitar a apenas explicar a configuração.

O SNORT possui um arquivo principal que contém as informações necessárias para que a máquina possa entender o que ela deve fazer, neste arquivo há configurações da rede e do arquivo que contem as regras e assinaturas de ataques.

O SNORT usa no arquivo snort.conf uma variável chamada HOME_NET e que serve para que não seja necessário especificá-la novamente em todos os arquivos de regras e

assinaturas de ataque, no arquivo snort.conf no entanto, é necessário que a variável HOME_NET seja associada ao IP da sua rede local.

Você também pode setar IP's para os servidores DNS ou para a variável EXTERNAL_NET, vai da necessidade de cada uma configurar da melhorar maneira possível, inclusive o próprio SNORT recomenda que para a variável EXTERNAL_NET seja associado o valor ANY, para que seja capturado tudo que passar por ela.

Pronto, configurado os IP's referentes a configuração da rede, a preocupação maior agora é de mostrar ao SNORT que arquivos de regras ele deve ler quando iniciar, arquivo este que foi criado conforme mostra a figura no começo do artigo.

Estes arquivos são especificados no final do arquivo snort.conf veja na Figura 3.1 o modelo e como são feitas as chamadas para tais arquivos:

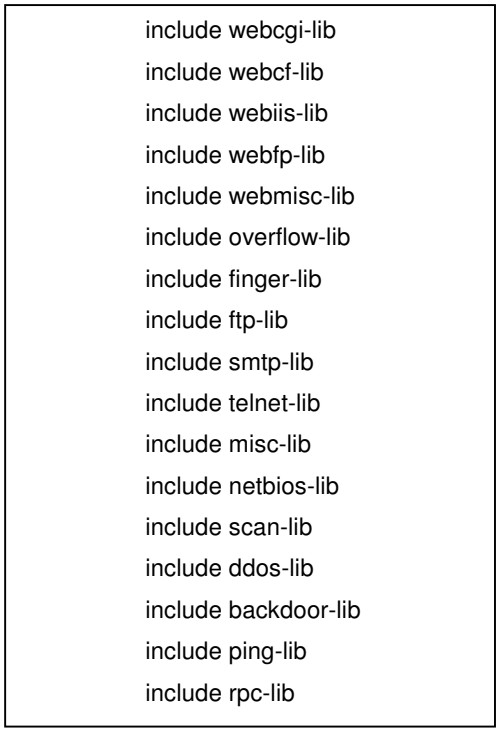


Figura 3.1 Arquivos de regra do SNORT

Vale lembrar que cada arquivo -lib contém informações sobre diferentes tipos de ataques, sendo assim quanto mais completa for a sua quantidade de arquivos, mais informações de ataques você receberá.

Após ter sido configurada as informações referentes a rede e ao IP's, acionaremos o snort através da seguinte linha de comando: snort -c /diretorio/snort.conf

## 3.4 Exemplo de funcionamento

Logo após a instalação e rodando normalmente, os log's já estão sendo gerados e a monitoração esta sendo efetuada, sendo assim podemos ter controle de tudo que está acontecendo verificando frequentemente os logs, abaixo segue um pequeno exemplo de como o SNORT reporta os ataques em seus logs(Figura 3.2):

```
[**] ICMP Destination Unreachable (Undefined Code!) [**]
04/04-03:17:10.137663 100.100.100.100 -> 200.200.200.200
ICMP TTL:245 TOS:0x0 ID:4939 IpLen:20 DgmLen:56
Type:3 Code:3 DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
200.200.200.200:1044 -> 100.100.100.100:53
UDP TTL:52 TOS:0x0 ID:25676 IpLen:20 DgmLen:91
Len: 51
** END OF DUMP
```

Figura 3.2 – Logs capturados com SNORT

No exemplo acima é possível verificar a abrangência das informações por ele especificadas, que nos trazem dados importantes como: origem, destino, ttl, tipo de ataque, horário, entre outros.

# CAPÍTULO 4 – O SISTEMA SADI

## 4.1 Estrutura

Toda a estrutura do projeto foi desenvolvida em linguagem C++, sobre a plataforma operacional Windows, utilizando a biblioteca Winpcap para captura de pacotes. Para os módulos visuais foi utilizada ambiente / ferramenta de programação Delphi 6/7. Para armazenar os pacotes capturados foi usado o banco de dados em Mysql.

Para fazer a conexão do banco com o Delphi, foi utilizado um componente, chamado ZEOS, que é de código livre. Este componente é de extrema importância para o projeto, pois o Delphi não tem suporte nativo para o banco MySQL.

Este projeto foi desenvolvido em vários módulos, sendo que no primeiro, foi implementada a captura e a interpretação dos pacotes, e esta informação é obtida hexadecimal, e para conseguir ler estes dados foi feita a converção de hexadecimal para ASCII.

### 4.2.1 Módulos

Foram desenvolvidos vários módulos para se chegar a uma plataforma visual, desde a captura em baixo nível até a interação com o usuário, mas de pouco funcionamento.

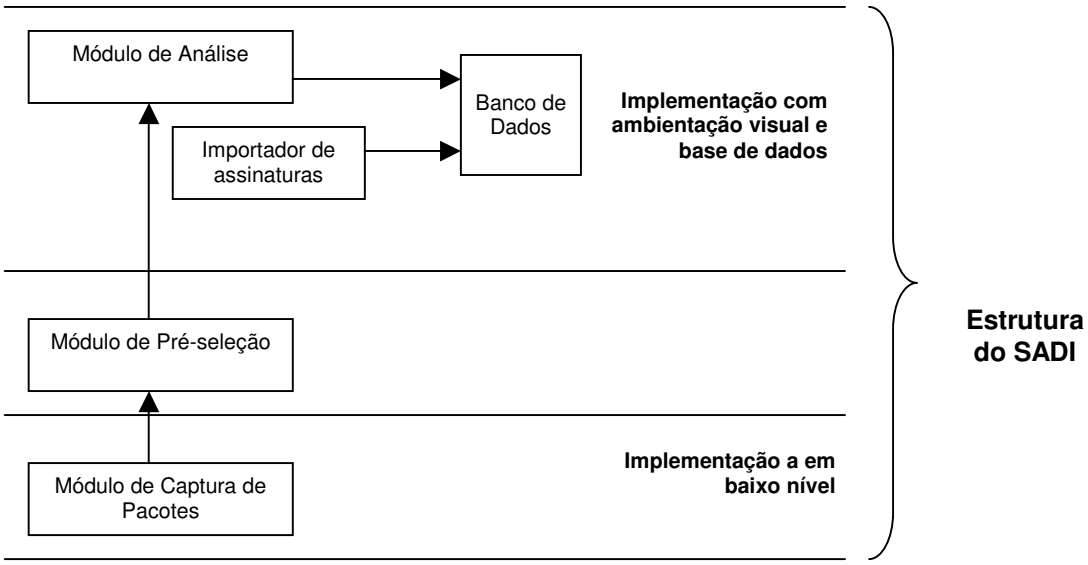


Figura 4.1: Esquema do SADI

### 4.2.2 Módulo de captura de pacotes

Foi desenvolvido para filtrar os pacotes em baixo nível (nível de Kernel) para não sobrecarregar os módulos seguintes. O filtro colocado no sistema para fins de teste permite a captura dos pacotes contendo os protocolos de transporte TCP e UDP. É um filtro que faz parte da biblioteca Winpcap e é muito eficiente.

Este módulo foi desenvolvido em linguagem C++, e para a captura e análise dos pacotes foi utilizada a biblioteca Winpcap (que será descrita melhor nos próximo capítulo).

#### 4.2.2.1 Biblioteca de Captura de Pacotes

Winpcap é uma biblioteca desenvolvida para capturar, filtrar e tratar os dados dos pacotes, e foi desenvolvida para rodar na plataforma Windows, mas há uma desenvolvida para rodar na plataforma Linux, chamada de Lipcap.

Para captura dos pacotes no projeto, foram utilizados três cabeçalhos, são eles, winsock2.h, remote-ext.h, pcap.h, e eles contêm as funções necessárias para se fazer a captura dos dados.

### 4.2.2.2 As funções principais da biblioteca Winpcap

Na estrutura pcap se encontra o descritor de captura, que é utilizado na maioria das funções.

A estrutura pcap_if contém duas funções muito importantes, a **pcap_findalldevs (),** que é utilizada para retornar os endereços dos dispositivos no ponteiro alldevs e pcap_freealldevs () que serve para liberar a lista de dispositivos. O uso da estrutura pcap_t também é obrigatório, já que é necessária para iniciar a captura dos pacotes, esta estrutura possui a função pcap_open, que retorna no ponteiro o nome do dispositivo de rede, o tamanho máximo do pacote que se deseja capturar, flag que define o estado do dispositivo, promiscuo ou não promiscuo, tempo de espera em milliseconds, e o valor null, este valor significa que estamos trabalhando localmente, valores diferentes que null, significa que iremos trabalhar remotamente e o buffer, que contem a mensagem de erro.

**Promiscuous vs. Non-promiscuous:** São duas técnicas de captura diferentes, a non-promiscuous apenas captura o trafego, de, para, ou roteado pelo host. Entretanto, está direcionado para ser utilizado em um Data link  Layer. O modo promiscuous, captura todo trafego do broadcast. Sua vantagem sobre o modo non-promiscuos é poder capturar um maior número de pacotes. (DIAS et al, 2004)

Para a recepção dos pacotes, é utilizada a função pcap_next_ex.

#### 4.2.2.3 Operações de captura

São definidas as variáveis para o nosso objetivo que é capturar o trafego que passa pela interface definida pelo usuário, depois de declarada as variáveis, entra em ação as funções, como explicado logo acima, no item 4.2.1.2 é utilizada uma função para procura automática de dispositivos de rede, pcap_findalldevs, depois de terminado este processo, com a placa setada em modo promiscuo, foi utilizada a função pcap_open para iniciar a captura dos pacotes, em seguida é utilizada a função pcap_next_ex para recepção dos pacotes, neste momento o pacote não esta decodificado, e é salva em um arquivo texto que posteriormente é decodificado para que se torne claro as informações contidas no pacote. (DIAS et al, 2004)

### 4.2.3 Módulo de pré–seleção

Foi necessário para que a interpretação do módulo de análise fosse mais fácil, tornando assim o trabalho mais ágil e rápido.

Logo após se capturar dos pacotes, é feita à sua análise, para poder se detectar uma intrusão, que é a principal função desse módulo.

Foram criadas três estruturas, uma estrutura para armazenar os endereços IPs, origem e destino, outra para armazenar o datagrama IP e outra para armazenar o datagrama UDP. Neste módulo pode ser apenas usado para detectar os datagramas UDP e TCP.

Após serem criadas as estruturas, foi criada uma função para a análise dos pacotes, Packet_handler, e tem como argumentos, um ponteiro de caracteres sem sinal, um ponteiro para a struct pcap_pkthdr, e outro ponteiro de caracteres.

Logo em seguida é feita a declaração obrigatória dos ponteiros. O programa é basicamente o mesmo da captura com a inclusão de novas funções que será comentada agora.

**pcap_compile(),** é usada para compilar expressões em filtros. Funciona desta maneira, o primeiro argumento é o descritor de captura, ponteiro *adhandle,* depois de especificado, a estrutura bpf_program compila a expressão passada como argumento, se for reconhecida como uma expressão se torna um filtro.

**pcap_setfilter(),** esta função é usada para setar o filtro compilado pela função pcap_compile(). O ponteiro adhandle recebe da estrutura bpf_program a expressão compilada em filtro e seta no kernel. Isto é importante frisar, setando no kernel, o programa ganha em desempenho.

**pcap_datalink(),** retorna o link layer type, na biblioteca <net/bpf> possui uma lista de códigos DLT_*, que são apropriados para plataformas compatíveis com o filtro.

**pcap_loop(),** com essa função é iniciada a captura dos pacotes e é determina quantos pacotes serão capturados. (DIAS et al, 2004)

### 4.2.4 Módulo de análise de pacotes

A principal função desse módulo é a identificação de qual protocolo que foi utilizado para o transporte do pacote capturado. A próxima etapa é a separação de todas as informações contidas nos cabeçalhos de cada protocolo de transporte.

### 4.2.5 Módulo importador de assinaturas

A função deste módulo é pegar arquivos de texto (exemplo: w11.txt) com as assinaturas de ataque e importá-las para o banco.

Estas informações serão utilizadas no módulo de análise para a comparação com os pacotes capturados.

## 4.3 Características gerais

O projeto tem todos os dados de cada campo do protocolo utilizado de cada pacote capturado, e separados para poderem ser utilizados. Os campos de dados são separados dos campos de cabeçalhos para aumentar o desempenho da análise. Após isso, a parte de dados do pacote está pronta para ser comparada com uma base de dados que será descrita a seguir. (DIAS et al, 2004)

## 4.4 Banco de dados

Foi estabelecido um banco de dados em MySQL (Open Source), que é uma peça fundamental para o projeto, pois é utilizado para a comparação dos comandos registrados no banco com os comandos decodificados no Sistema, conseguindo desta maneira detectar um comando perigoso contido em um pacote, o que caracteriza um intruso.

## 4.5 Utilização dos dados capturados para a detecção de intrusos

Estes dados que foram capturados servem apenas para detecção e não estão sendo armazenados no banco, servem apenas para a tomada de decisão pelo administrador de rede baseado em uma detecção de intrusão, mas poderão vir a ser útil na detecção de problemas na rede, como gargalo, lentidão, e outros. (DIAS et al, 2004).

# CAPÍTULO 5 – O DESENVOLVIMENTO DE UM MÓDULO DE ACESSO REMOTO E DE NÍVEIS DE SEGURANÇA

## 5.1 Introdução

Para melhorar o desempenho do SADI, foi desenvolvidos 2 módulos extras, e foram feitas algumas alterações no módulo anterior, para melhorar a interface visual.

Para deixar o SADI ainda mais completo foi necessário acrescentar, ao módulo de análise do projeto anterior, um novo módulo, que tem a função de analisar e classificar os ataques em Níveis de Segurança, e armazena-los para que possam ser vistos pelo usuário do sistema.

E para poder visualizar todas as informações que contém o SADI, um novo módulo, que também foi adicionado ao projeto, foi o Módulo de Acesso Remoto, no qual o usuário do SADI pode acessar de qualquer lugar do planeta através de um navegador de internet o endereço do mesmo, e ter informações atuais que estão sendo capturadas pelo SADI.

## 5.2 Objetivos

Este projeto tem como o objetivo principal o aperfeiçoamento do Sistema de Detecção de intrusão (SADI) através da adição da capacidade de análise dos pacotes capturados, procurando-se especificar o grau de periculosidade do acesso. Também faz parte deste aperfeiçoamento, a criação de um serviço de acesso remoto, para a configuração e monitoramento, através de uma interface gráfica com o usuário.

## 5.3 Estrutura do Projeto

Foram estabelecidos três módulos, localizados no topo da estrutura hierárquica do SADI para a continuidade do projeto SADI, são elas:

- Módulo de análise de níveis de segurança.

- Módulo de Acesso Remoto.

- Módulo Principal de Interface gráfica.

Cada um destes módulos será descrito nos capítulos a seguir e podemos ver ,onde entram na arquitetura do SADI conforme Figura 5.1 .

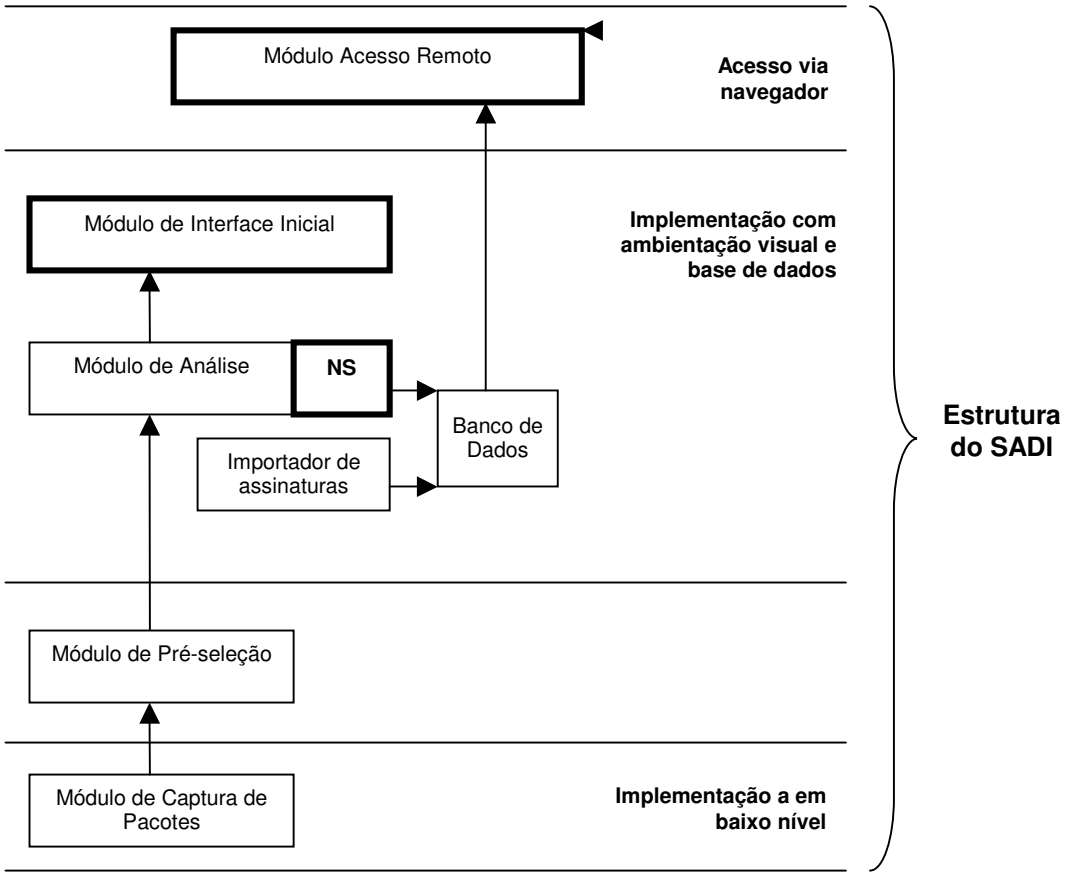


Figura 5.1: Arquitetura do SADI

### 5.3.1 Módulo de Acesso Remoto

Este módulo foi desenvolvido totalmente sobre plataformas livre, uma vez que o banco de dados que está sendo utilizado (Mysql), e a implementação foi feita em PHP.

Uma das principais características desse módulo é a facilidade de uso, e com uma interface gráfica simples e amigável ao usuário.

O módulo de acesso remoto é acessado através de qualquer computador, através de um navegador de internet. É necessário que seja feita uma autenticação do usuário no sistema, antes que ele possa acessar o mesmo, conforme Figura 5.2. Ele deve ser instalado no mesmo ponto estratégico que o programa, mas será necessário um servidor de internet, no caso foi utilizado o AppServ, que é uma plataforma livre.


Figura 5.2: Tela de autenticação de usuário do Módulo de Acesso Remoto

### 5.3.1.1 Banco de dados - Tabela de usuário

Foi adiciona no banco a tabela usuário e criada uma tela de autenticação de usuário (logon), para poder acessar o módulo de acesso remoto, visando a segurança dos arquivos, para que não seja acessado por pessoal não autorizado, conforme podemos ver na figura acima.

Na tabela usuário, estão os seguintes campos, conforme Tabela 5.1:

Tabela 5.1 - Dados da tabela de usuário

| **CAMPO** | **TIPO** | **DESCRIÇÃO** |
|---|---|---|
| *CODIGO* | INT (6) | que é o código de usuário e usado apenas para controle do banco |
| *NOME* | VARCHAR (20) | nome do usuário é utilizado para saudação ao usuário, pois este módulo é voltado ao mesmo |
| *LOGIN* | VARCHAR (10) | o login escolhido pelo usuário do sistema SADI |
| *SENHA* | VARCHAR (10) | senha que é privada a cada usuário |
| *NIVEL* | INT 3 | que é nível de autorização), que tem diferentes privilégios para o usuário, são elas:<br>**1:** para consultar, alterar e excluir dados, e poder adicionar novos usuários e visualizar suas respectivas senhas;<br>**2:** para apenas poder consultar os dados. |
| *DATA* | DATE | é armazenada a data que o usuário acessou o sistema em sua ultima visita |
| *HORA* | TIME | é armazenada a hora que o usuário acessou o sistema em sua ultima visita ao módulo |

### 5.3.1.2 Tela Principal e todas suas funções

Foi criada uma tela com interação total com o usuário do sistema de acesso remoto, na qual, contém informações importantes, como o número de assinaturas que estão armazenados até o presente momento no banco de dados do sistema, quantos usuários tem armazenado, o total de intrusões já foram detectadas pelo SADI.

Há algumas informações importantes apresentadas pelo sistema, como a data e a hora do ultimo acesso do mesmo usuário e quantos ataques foram detectados pelo SADI, desde o ultimo acesso.

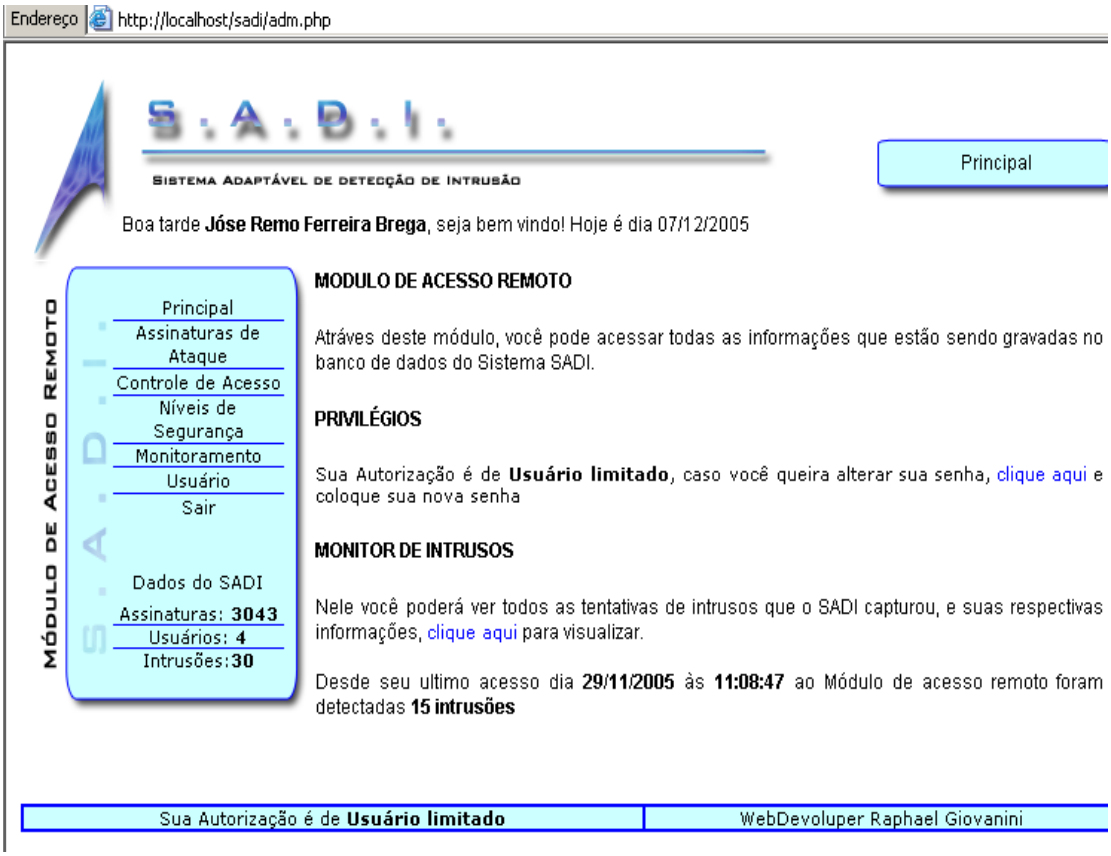


Figura 5.3: Tela principal do módulo de acesso remoto

Algumas das funções disponíveis nesse módulo são:

**Alterar Senha:** nesta opção pode ser alterada a senha do usuário atual do sistema. Esta localizada apenas na tela principal

**Assinaturas de Ataque:** lista todas as assinaturas de ataque registradas no banco de dados até o momento presente. Conforme Figura 5.4.

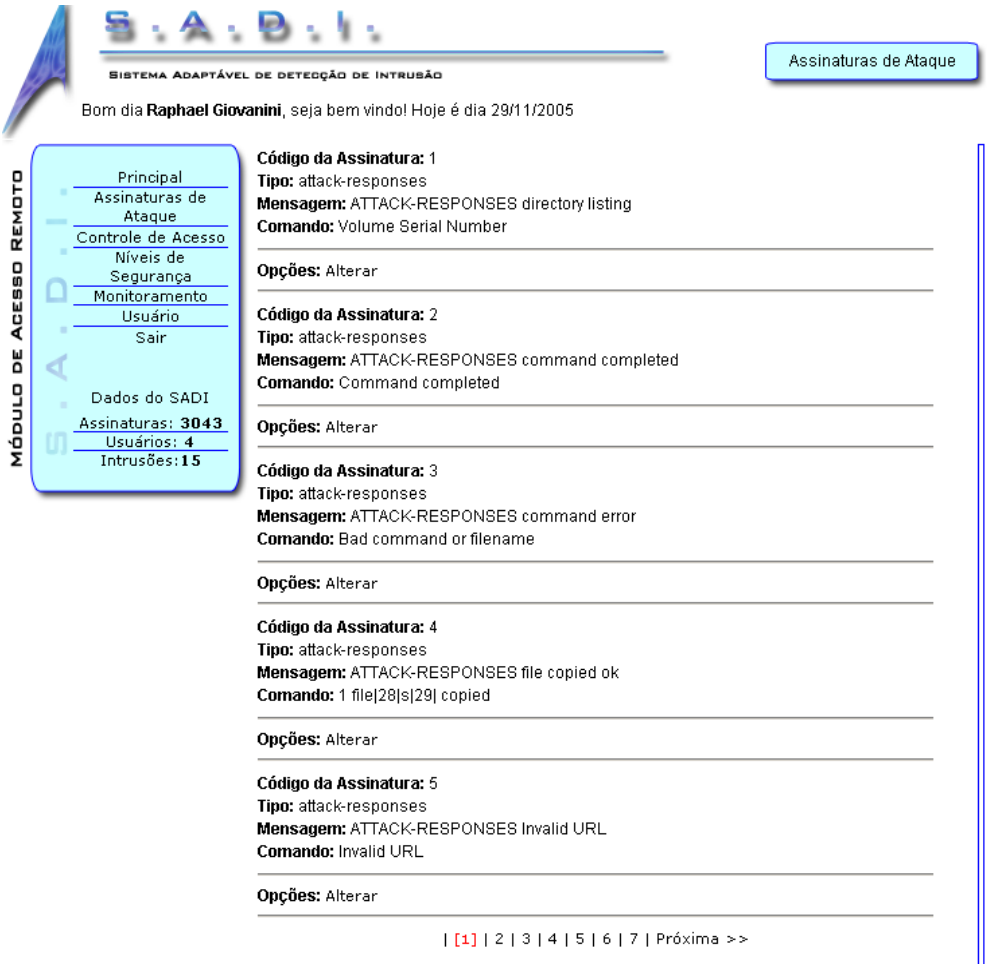


Figura 5.4: Assinaturas de Ataque

**Controle de Acessos:** mostra todos os usuários que acessaram o sistema, mostrando data, hora.

**Níveis de Segurança:** lista os níveis de segurança registrados no banco de dados até o presente momento. Caso o usuário seja administrador do sistema ele pode visualizar as

senhas de todos usuários do sistema como pode alterar qualquer informação e a inclusão de novos usuários.

**Usuário:** lista os usuários do sistema. Caso seja com autorização de Administrador, como descrito no banco de dados de usuário do SADI, pode visualizar a senha de todos os usuários e também pode incluir novos usuários.

**Monitoramento:** visualização de todos os ataques que foram armazenados no SADI, em seu módulo principal até o presente momento. O interresante é modo que é tratado, pois quando a intrusão é classifica por cores: verde para baixo risco, amarelo pra risco médio e vermelho pra alto risco.

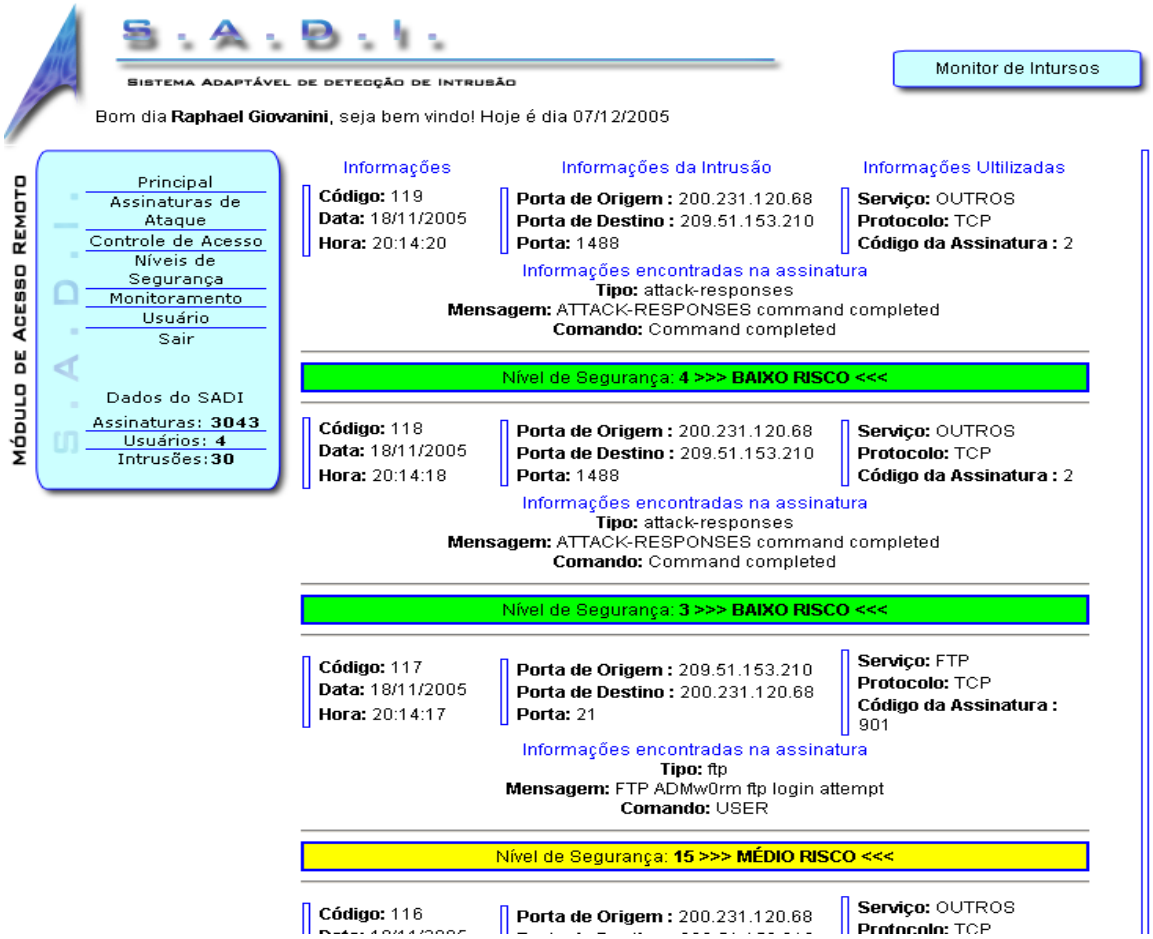


Figura 5.5 – Informações de monitoramento

### 5.3.1.3 Detalhes importantes do sistema

É importante ressaltar, que por estarmos abordando o assunto segurança, este módulo foi desenvolvido com um código seguro, pois há uma necessidade de se saber quando ele esta sendo usado e por quem, então foi criado no banco de dados uma tabela na qual armazena os dados contidos na Tabela 5.2:

Tabela 5.2 – Dados da tabela de Logs

| **CAMPO** | **TIPO** | **DESCRIÇÃO** |
|---|---|---|
| *CODIGO* | INT (6) | o número atual de acesso, em relação a todos os usuários |
| *DATA* | DATE | a data em que está sendo acessado o sistema |
| *HORA* | TIME | a hora em que está sendo acessado o sistema |
| *CODUSUARIO* | INT (3) | o código do usuário que está acessando o sistema |
| *ALTERACAO* | VARCHAR (10) | que é o procedimento de logar no sistema naquele momento |

Antes da inserção é verificado se existe um registro e qual foi seu ultimo do usuário que está efetuando a entrada no módulo de acesso remoto, através do código PHP da Figura 5.6:

```php
//PARA SABER O CÓDIGO DE USUÁRIO
$sql = "SELECT codigo FROM usuario WHERE login = '$usuario'";
$codigo = mysql_query($sql) or die(mysql_error());
$cli = mysql_fetch_array($codigo);
$codigo= $cli['codigo'];

//ARMAZENA ULTIMA DIA E HORARIO QUE O SISTEMA FOI ACESSADO
 $consulta_ultima = "SELECT MAX(codigo),data,hora FROM log
            WHERE codusuario='$codigo'
            GROUP BY data,hora
            ORDER BY codigo DESC
            ";
$visita = mysql_query($consulta_ultima) or die(mysql_error());
if($visita){
$uv = mysql_fetch_array($visita);
$ultima_hora = $uv['hora'];
$ultima_data = $uv['data'];

$sql = "UPDATE usuario
          SET
          data ='$ultima_data',
          hora = '$ultima_hora'
          WHERE login ='$usuario'";
$altera=mysql_query($sql);
}else{//SE ELE ACESSA PELA PRIMEIRA VEZ
 $sql = "UPDATE usuario
          SET
          data ='$data',
          hora = '$tempo'
          WHERE login ='$usuario'";
$altera=mysql_query($sql);
}
$alteracao='Logar no sistema';
$sql = "INSERT INTO log
           (data,hora,codusuario,alteracao)
            VALUES ('$data','$tempo','$codigo','$alteracao')";
$insere = mysql_query($sql) or die(mysql_error());
//FIM DO LOG DE CONTROLE
```

Figura 5.6: Trecho de Código do Controle de Acesso

Na tela *logon.php* é digitado o usuário e a senha, esses dados são passados para *valida_logar.php*, na qual é feita a verificação se o usuário existe, caso não exista irá para uma tela acusando erro, caso o usuário exista a variável em que foi digitada o usuário *$usuario*, é utilizada em um SELECT com finalidade de se pegar o código de usuário em uma variável

### 5.3.2 Módulo de análise de níveis de segurança

Este módulo foi adicionado ao módulo de pré-seleção do SADI, em que utilizando das informações da intrusão que foi realizada, mais os dados contidos na tabela de NIVEIS, e

gera um novo registro no banco do SADI classificando o risco e salva na tabela MONITORAMENTO, que pode ser usada no módulo de acesso remoto.

Para esse módulo de logo de início foi estabelecido o nível de segurança (NS), que é um valor numérico que vai sendo acrescido à medida que são monitorados os eventos de interesse provenientes de uma dada origem. (SOUSA, FILHO, 1997, p 21).

Logo após a identificação de um pacote de intrusão é comparado com a tabela de NS, que foi adicionada ao banco do projeto e a tabela de monitoramento.

Um exemplo para ilustração do uso do NS: na rede que está sendo monitorada é detectado um *finger* vindo do *host* externo *A* para o *host B, e que* inclui um vetor para este evento na lista de monitoração, com um valor de nível de segurança pré-determinado para o *finger* (NS = 2). Ao detectar um novo evento, por exemplo, um *telnet* (NS = 15), é consultada a tabela *monitoramento* para saber se a entrada provém da mesma origem-destino, e o nível de segurança deste vetor existente é acrescido do valor do nível de segurança do *telnet*, ficando com NS = 17. Neste caso, não é criado um novo vetor na fila. (SOUSA, FILHO, 1997, p 21)

Pode acontecer também outra situação em que a intrusão venha da mesma origem de um que já foi registrado, mas não necessariamente vindo da mesma máquina, nem possivelmente com mesmo destino dentro da rede sob monitoramento. Neste caso um novo vetor é inserido na tabela monitoramento, mas com o nível de segurança equivalente, acrescido de um peso que indica que já há alguma ocorrência vinda daquele domínio. Outras variações podem ser facilmente expandidas, conforme necessário, a partir das duas situações principais acima. (SOUSA, FILHO, 1997, p. 22)

O tratamento do Nível de Segurança segue as especificações apresentadas no modelo de detecção. Ele é incrementado à medida que são monitorados eventos de interesse. Os valores dos Níveis de Segurança definidos no projeto para cada um dos possíveis eventos, serviços de rede, está listado na Tabela 5.3.

Tabela 5.3 - Valores de níveis de segurança, referentes às portas, para determinar o disparo automático de captura.

| Protocolo | Porta | Nível de Segurança | Serviço |
|---|---|---|---|
| TCP | 21 | 15 | FTP |
| TCP | 23 | 15 | Telnet |
| TCP | 25 | 5 | SMTP |
| UDP | 67 | 10 | Bootp |
| TCP | 79 | 2 | Finger |
| TCP | 110 | 5 | Pop3 |
| UDP | 11 | 10 | SunRPC |
| TCP | 512 | 5 | Exec |
| UDP | 513 | 10 | Rlogin |
| TCP | 514 | 15 | Rsh |
| TCP | 515 | 10 | Lpr |
| UDP | 520 | 10 | Route |
| UDP | 2049 | 15 | NFS |
| Outras | qualquer | 1 | Outras portas não listadas (default) |

Estes valores foram calculados baseados nas capacidades de cada serviço. O *telnet,* por exemplo, oferece mais capacidade (e, portanto maior risco de intrusão) do que o *finger*, portanto aquele terá um nível de segurança maior do que este. Os valores de TS e TL são importantes para se proceder a uma análise periódica e cíclica da tabela. Assim certas conexões serão removidas após um determinado tempo de vida, uma vez que se decida que elas não são mais relevantes para a análise de intrusão.

Baseado nos Níveis de Segurança foi definido um valor máximo para o NS, o valor de 30, e caso seja ultrapassado é considerado de alto risco e o *host* e o domínio ao qual este pertence, passa a ser considerado potencialmente perigoso para a rede.

É considerado de médio risco valores de NS acima de 10, abaixo disso são considerados de baixo risco.

Sendo assim, foi adicionada no módulo de análise a seguinte trecho de código, logo após ter sido feita a captura das assinaturas de ataque, conforme Figura 5.7 e 5.8:

```
//inserçao na tabela monitoramento
qryMonitoramento.Insert;
qryMonitoramentoorigem.AsString := DM.Qry_CabecalhoIP_Origem.AsString;
qryMonitoramentodestino.AsString := DM.Qry_CabecalhoIP_Destino.AsString;
qryMonitoramentodata.AsDateTime := Date;
qryMonitoramentohora.AsDateTime := Time;
qryMonitoramentoporta.asString := DM.Qry_CabecalhoPorta_Origem.AsString;
if ( dm.Qry_CabecalhoProtocolo.asInteger = 6 ) then
        qryMonitoramentoprotocolo.asString := 'TCP';
if ( dm.Qry_CabecalhoProtocolo.asInteger <> 6 ) then
        qryMonitoramentoprotocolo.asString := 'UDP';
//aqui vamos buscar os valores ns e servico da tabela niveis
qrySelecao.Open;
qrySelecao.SQL.Clear;
qrySelecao.SQL.Add('SELECT niveis.ns AS ns, niveis.servico AS servico');
qrySelecao.SQL.Add('   FROM niveis  ');
qrySelecao.SQL.Add('  WHERE niveis.protocolo = '' + qryMonitoramentoprotocolo.asString + '');
qrySelecao.SQL.Add('    AND niveis.porta = ' + qryMonitoramentoporta.asString);
qrySelecao.Open;
if ( qrySelecao.fieldbyName('servico').IsNull ) then
   begin
      servico := 'OUTROS';
      nivel_seguranca := 1;
    end
```

Figura 5.7: Trecho de código 1

```
      else
          begin
              servico := qrySelecao.fieldbyName('servico').AsString;
              nivel_seguranca := qrySelecao.fieldByName('ns').AsInteger;
           end;
      //neste ponto a query ja fez a pesquisa e os dados estao carregados..
      qryMonitoramentons.asInteger := nivel_seguranca;
      qryMonitoramentoservico.asString := servico ;

      qryPesquisa.Close;
       qryPesquisa.SQL.Clear;
       qryPesquisa.SQL.Add('SELECT MAX(ns) AS ns FROM monitoramento where origem = '''
+ qryMonitoramentoorigem.asString + '''');
      qryPesquisa.SQL.Add('   AND destino =  ''' + qryMonitoramentodestino.asString + '''');
      qryPesquisa.Open;
      if not (qryPesquisa.IsEmpty ) then
          qryMonitoramentons.AsInteger := qryMonitoramentons.asInteger +
qryPesquisa.fieldByname('ns').asInteger ;

       qryMonitoramento.Post;
 qryMonitoramento.CommitUpdates;

      Dm.Qry_Risco.Next;
```

Figura 5.8 – Trecho de Código 2

### 5.3.3 Banco de Dados de NS – Tabela NIVEIS e MONITORAMENTO

Esta tabela foi inserida no banco de dados do SADI e é chamada de NIVEIS, em que são armazenados os dados apresentados na Tabela 5.3 e possui os campos conforme Tabela 5.4:

Tabela 5.4 – Dados da Tabela de Níveis

| **CAMPO** | **TIPO** | **DESCRIÇÃO** |
|---|---|---|
| *CODIGO* | INT (3) | É o código da tabela de NIVEIS |
| *PROTOCOLO* | VARCHAR (20) | É o protocolo usado |
| *PORTA* | VARCHAR (5) | É a porta de destino pela qual é feita a invasão |
| *NS* | INT (3) | Um valor atribuído conforme a tabela 5.3 |
| *SERVICO* | VARCHAR (20) | É serviço utilizado para o ataque |

Esta tabela também foi inserida no banco de dados do SADI e é chamada de MONITORAMENTO, em que são armazenados os dados de ataque mais os campos da tabela Níveis, conforme Tabela 5.5.

Tabela 5.5 – Dados da Tabela de Monitoramento

| **CAMPO** | **TIPO** | **DESCRIÇÃO** |
|---|---|---|
| *CODIGO* | INT (3) | É para controle |
| *ORIGEM* | VARCHAR (16) | É armazenado o IP de origem |
| *DESTINO* | VARCHAR (16) | É armazenado o IP de destino |
| *DATA* | DATE | Data em que foi feito o ataque |
| *HORA* | TIME | Em que foi feito o ataque |
| *PROTOCOLO* | VARCHAR(20) | É equivalente ao da tabela de NIVEIS |
| *PORTA* | VARCHAR(5) | É equivalente ao da tabela de NIVEIS |
| *NS* | INT(3) | É equivalente ao da tabela de níveis |
| *SERVICO* | VARCHAR(20) | É equivalente ao da tabela de NIVEIS |
| *CODASS* | INT (3) | O campo em que é armazenado o código da assinatura que foi detectada, que tem um campo correspondente na tabela do banco de dados de assinatura chamada de código. |

# CAPÍTULO 6 - Testes

## 6.1 Ambiente experimental

Os testes foram realizados em um computador Athlon XP 2Ghz, 256MB, placa de rede onboard

## 6.2 Testes Realizados

Foram realizados testes nos novos módulos, o de acesso remoto e o de Nível de Segurança, os testes realizados.

Como não houve necessidade de simular ataques, pois no computador em que está rodando a aplicação, é bombardeado de ataques.

### 6.2.1 Testes no Módulo de Acesso Remoto

Foram realizados testes de acesso de máquinas remotas, na máquina em que foi colocado em ponto estratégico da rede para capturar todos os pacotes, e nela foi instalado também, o módulo de acesso remoto, que foi acessado através do endereço http://*200.231.120.68/sadi/logar.php.*

Acessado todas as suas funções com dois usuários que foram criados, um com permissão de administrador do sistema e outro com autorização de usuário simples.

Ao ser capturado um ataque o sistema automaticamente grava os dados do mesmo na tabela de monitoramento, que pode ser acessada pelo módulo de acesso remoto.

### 6.2.2 Teste no SADI com o módulo de análise de Nível de Segurança

Foi testado o novo módulo de análise com a parte de nível de segurança, depois de alguns minutos rodando, todos os pacotes que foram capturados foram considerados de risco, e foram corretamente adicionados a tabela.

O teste foi realizado dia 18 de novembro às 20 horas da noite e a aplicação foi rodada por cerca de um minuto e meio, devido a grande quantidade de pacotes de risco que foi capturada.

Essa grande quantidade de pacotes de risco se deve ao fato, de que o servidor de internet ao qual o computador está utilizando, estar contaminada com vários worms e vírus.

Foram detectadas 15 assinaturas de ataques das quais as muitas delas vieram do mesmo local e não usou nenhum dos protocolos e portas da tabela de nível de segurança, mas apenas um, através do protocolo TCP, usando a porta 21 e através do FTP, tendo um nível de segurança igual a 15, considerado de risco médio.

Veja o resultado na Figura 6.1.

| cod | origem | destino | data | hora | protocolo | porta | ns | servico |
|---|---|---|---|---|---|---|---|---|
| 105 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 1 | OUTROS |
| 106 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 2 | OUTROS |
| 107 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 3 | OUTROS |
| 108 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 4 | OUTROS |
| 109 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 5 | OUTROS |
| 110 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 6 | OUTROS |
| 111 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 7 | OUTROS |
| 112 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 8 | OUTROS |
| 113 | 207.46.7.8 | 200.231.120.68 | 2005-11-18 | 20:14:04 | TCP | 80 | 9 | OUTROS |
| 114 | 200.231.120.68 | 207.46.7.8 | 2005-11-18 | 20:14:05 | TCP | 1055 | 1 | OUTROS |
| 115 | 200.231.120.68 | 209.51.153.210 | 2005-11-18 | 20:14:15 | TCP | 1488 | 1 | OUTROS |
| 116 | 200.231.120.68 | 209.51.153.210 | 2005-11-18 | 20:14:16 | TCP | 1488 | 2 | OUTROS |
| 117 | 209.51.153.210 | 200.231.120.68 | 2005-11-18 | 20:14:17 | TCP | 21 | 15 | FTP |
| 118 | 200.231.120.68 | 209.51.153.210 | 2005-11-18 | 20:14:18 | TCP | 1488 | 3 | OUTROS |
| 119 | 200.231.120.68 | 209.51.153.210 | 2005-11-18 | 20:14:20 | TCP | 1488 | 4 | OUTROS |

Caso 1

Caso 2

Figura 6.1: Dados Capturados pelo SADI e armazenados no banco de dados.

**Caso1:** podemos observar que o IP de origem 2007.46.7.8 efetuou 9 tentativas de intrusão seguidas ao computador na qual está rodando o SADI e que por virem do mesma origem seu nível de segurança foi sendo aumentando com o valor do ataque atual mais o anterior, para se saber o risco atual.

**Caso2:** também ocorreu um ataque logo após se utilizar um programa de FTP, na maquina onde está o SADI, e com a tentativa de acesso foi capturada uma tentativa de ataque de nível de segurança 15, da seguinte origem 209.51.153.210.

# CAPÍTULO 7 – CONCLUSÕES, TRABALHOS FUTUROS E DIFICULDADES

## 7.1 Conclusões

Chega-se a conclusão que a ferramenta SADI com seus novos recursos se tornou uma ferramenta ainda mais poderosa e com totalmente planejada para interação com o usuário do mesmo. É uma ferramenta de detecção de intrusos que se tornou ainda mais eficiente após a adição do módulo de níveis de segurança, tornando assim os dados capturados disponíveis e com seu grau de periculosidade, para que o usuário através de um módulo de acesso remoto pode ser acessado todas as informações de base mais as coletadas pela ferramenta. Esta ferramenta é totalmente adaptável, podendo ser utilizada no estudo como ferramenta didática, como pode ser usada comercialmente.

## 7.2 Para trabalhos futuros

Algumas idéias para futuras implementações serão listas, como reconhecimento de mais protocolos, aprimoramento do nível de segurança, aprimoramento do acesso do módulo de acesso remoto, bloqueio do pacote considerado intruso.

### 7.2.1 Reconhecimento de protocolos.

Até o presente momento a ferramenta só identificar os protocolos TCP, UDP e IP, mas serem implementados ICMP, ARP, e outros.

### 7.2.2 Aprimoramento do nível de segurança

Pode ser feita através de rede neural, treinando a rede conforme ela é alimentada, pois o atual algoritmo não está treinando a rede, está apenas usando alguns valores adotados como padrão inicial.

### 7.2.3 Aprimoramento do módulo de acesso remoto

Colocar mais funções como acessar em tempo real o programa ou se estiver no módulo de acesso remoto assim que ocorrer um ataque ser avisado também neste módulo

### 7.2.4 Aprimoramento do módulo visual do SADI

Reorganizar todo o conteúdo para cada tipo de protocolo, e não de forma genérica.

### 7.2.5 Bloqueio de pacote intruso

O sistema somente avisa que foi capturado, ele apenas faz a análise, a idéia então seria poder tratar o pacote capturado e tratá-lo, como por exemplo, bloqueá-lo para analise ou a exclusão do mesmo.

## 7.3 Dificuldades encontradas no trabalho

A maior dificuldade encontrada nesse trabalho foi reutilizar o projeto desenvolvido anteriormente a este trabalho, pois havia total falta de informação no material disponibilizado, sendo que faltavam componentes usados e o banco de dados (dados e o sistema gerenciador).

### 7.3.1 Componentes visuais

No cd que foi disponibilizado com o SADI, não continham dois componentes usados em Delphi, são eles:

ZEOS: que é a conexão do Delphi com o sistema gerenciador de dados que foi feito em MySQL

TBSpeedButton: componente pra visual pra mudar forma de botões e sem utilidade nenhuma para o projeto.

### 7.3.2 Banco de Dados

Não foi disponibilizado o sistema gerenciador do banco de dados usado no projeto, MYSQL, e o banco do projeto e seus dados. Isto atrasou em muito o andamento do projeto, pois sem o banco original não poderia ter criado as tabelas que seriam usadas no sistema de acesso remoto.

# REFERÊNCIAS

CANSIAN, Adriano Mauro. **Desenvolvimento de Um Sistema Adaptativo de Detecção de Intrusos em Redes de Computadores**. Tese apresentada ao Instituto de Física de São Carlos, da Universidade de São Paulo, para obtenção do título de Doutor em Física Aplicada, sub área Física Computacional, São Carlos, 1997.

DIAS, Álann Carlos Monteiro, et al. **SADI – Sistema Adaptável de Detecção de Intrusão**. Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) – Centro Universitário de Marília, Fundação de Ensino Eurípedes Soares da Rocha, Marília, 2004.

SOUSA, Aleck Zander Tomé de; FILHO Sérgio Antônio Leugi. **Um sistema de Captura de Pacotes para Uso em Segurança de Redes**. Projeto Final de Curso (Graduação em Ciência da Computação) – UNESP Campus de São José do Rio Preto, São José do Rio Preto, 1997.

SILVA, Artur Renato A. **Um modelo representativo de assinaturas de ataque para sistemas detectores de intrusão**. Projeto Final de Curso (Graduação em Ciência da Computação) – UNESP Campus de São José do Rio Preto, São José do Rio Preto, 2002.

CAFFARO, Marcelo Leão. **Sistemas de Detecção de Intrusos**, mai. 2001. Disponível em: <http://www.securenet.com.br/artigo.php?artigo=95> Acesso em: 20 out. 2005.

WINPCAP. Biblioteca de captura de pacotes e toda sua documentação. Disponível em: <http://winpcap.polito.it>. Acesso em: 25 mar. 2005.

APPSERV. Servidor PHP e MySQL, foi usada a versão appserv-win32-2.4.0, site do desenvolvedor. Disponível em: <http://www.appservnetwork.com>, site onde. Acessado em: 01 fev. 2005.